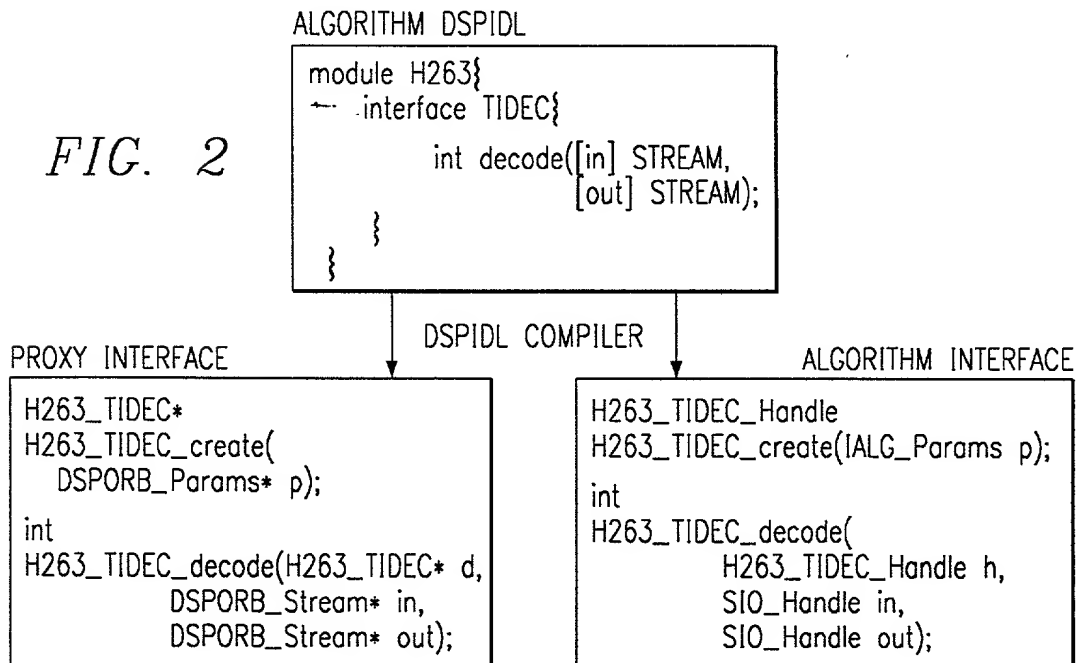
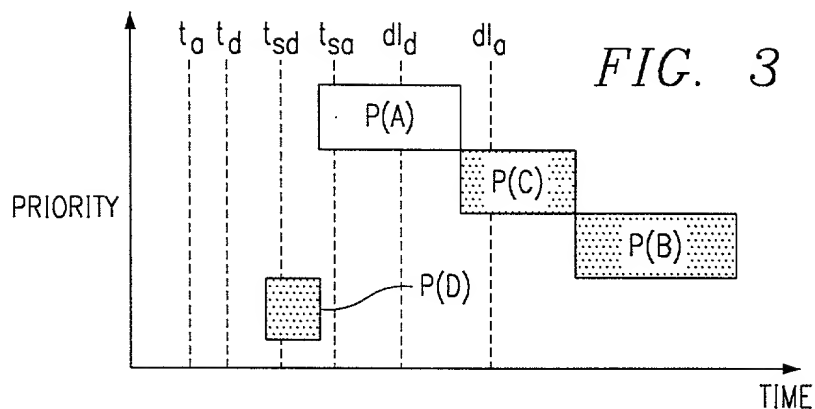


FIG. 1

→ DATA TRANSFERRED
 ----> DATA NOT TRANSFERRED

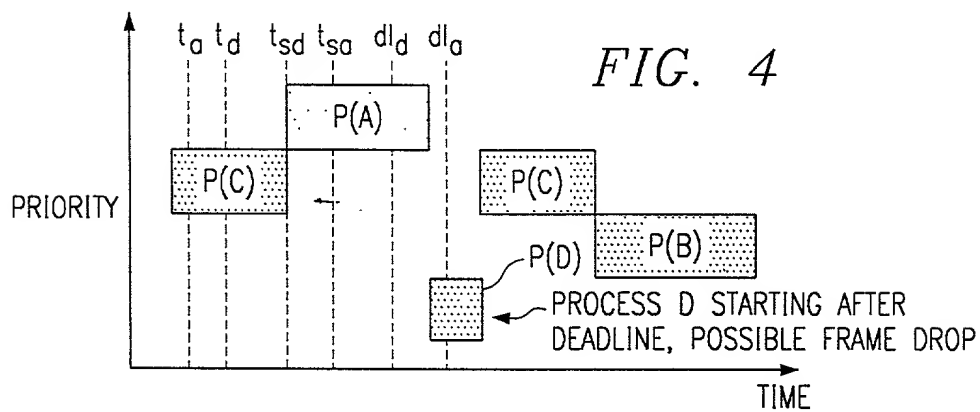
FIG. 2





t_{sa} = LAST POSSIBLE TIME FOR PROCESS A
TO START AND STILL MAKES ITS DEADLINE

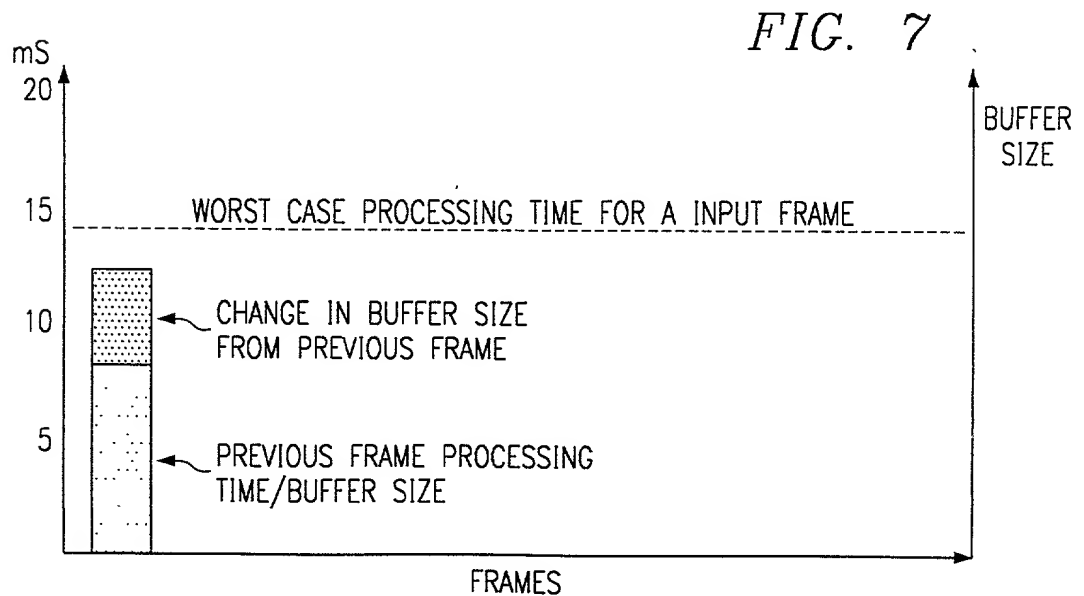
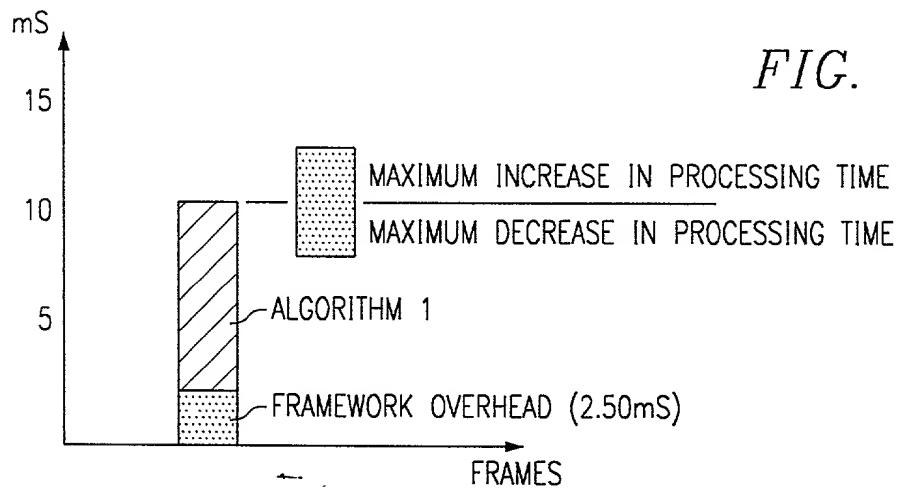
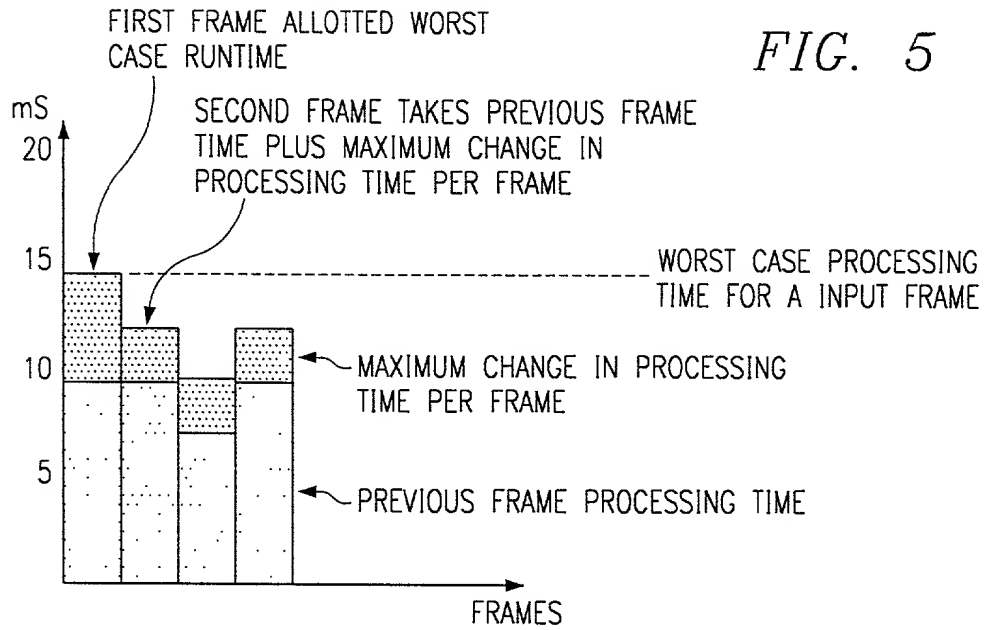
t_{sd} = LAST POSSIBLE TIME FOR PROCESS D
TO START AND STILL MAKE ITS DEADLINE



t_{sa} = LAST POSSIBLE TIME FOR PROCESS A
TO START AND STILL MAKES ITS DEADLINE

t_{sd} = LAST POSSIBLE TIME FOR PROCESS D
TO START AND STILL MAKE ITS DEADLINE

FOR "SECRET" 0941347 04360



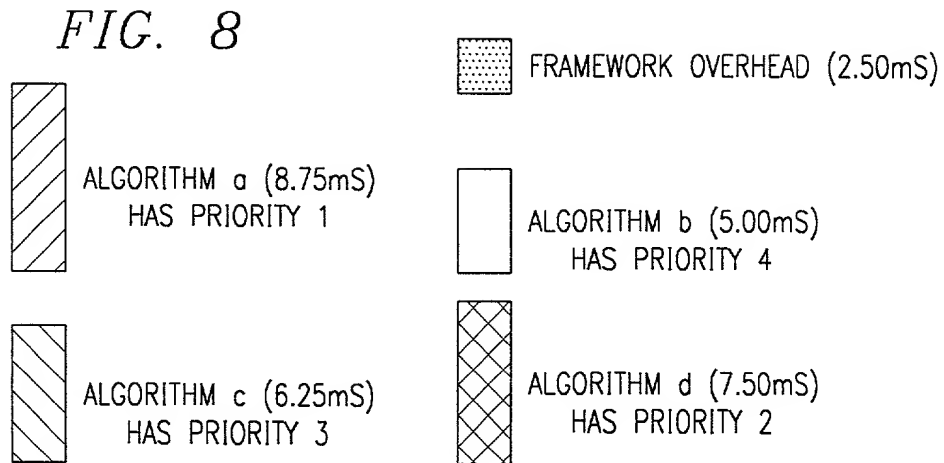
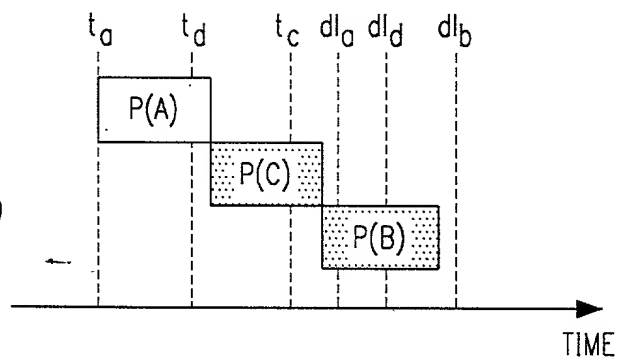


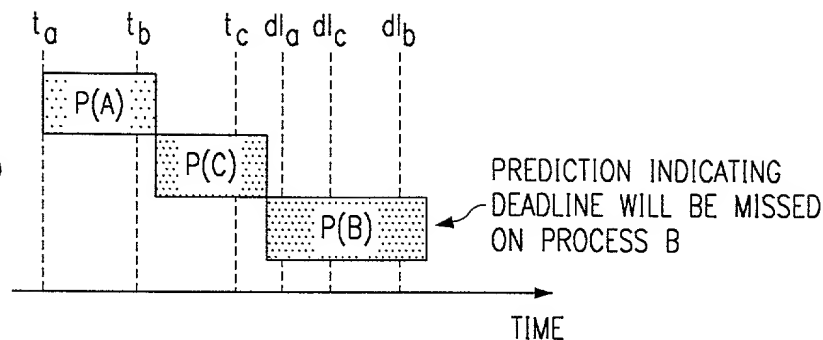
FIG. 9



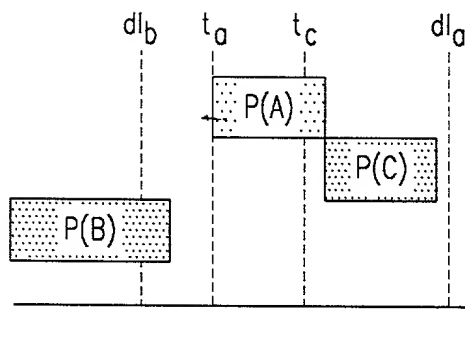
- t_i = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS
 dl_i = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME
 $P()$ = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME

205240" 242T4360

FIG. 10



- t_i = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS
- d_{li} = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME
- $P()$ = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME



BOTH PROCESS A AND C ARE PREDICTED TO COMPLETE BEFORE THEIR RESPECTIVE DEADLINES MEANING PROCESS B MISSING ITS DEADLINE DOES NOT RIPPLE THROUGH THE SYSTEM (YET)

- t_i = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS
- d_{li} = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME
- $P()$ = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME

FIG. 11

09341347.04301
T05340" 443T4850

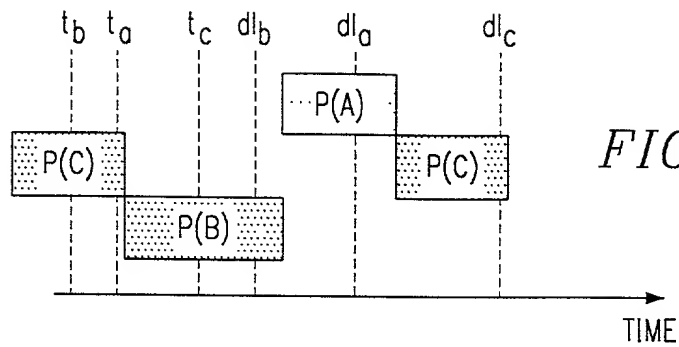


FIG. 12

- t_i = TIME STAMP ARRIVAL OF EACH DATA
FRAME FOR THE RESPECTIVE PROCESS
- dl_i = DEADLINE FOR FINISHING PROCESSING
OF EACH RECEIVED DATA FRAME
- $P()$ = PREDICTION OF PROCESSING TIME
FOR EACH RECEIVED DATA FRAME

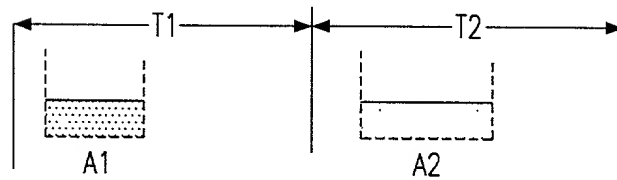


FIG. 13a

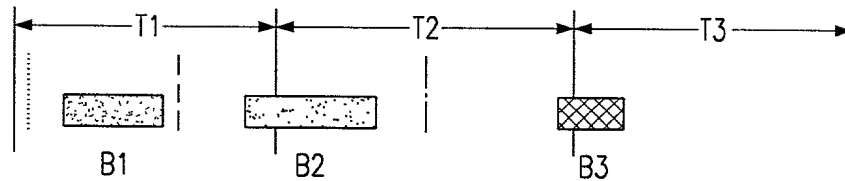


FIG. 13b

- ARRIVAL OF BUFFER B1
- ARRIVAL OF BUFFER B2
- ARRIVAL OF BUFFER B3

FIG. 14

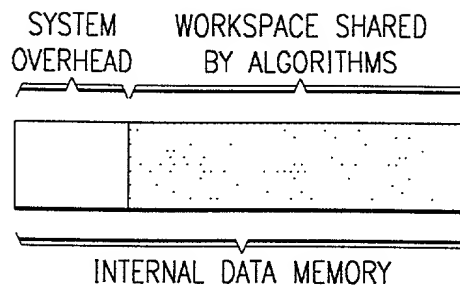


FIG. 15

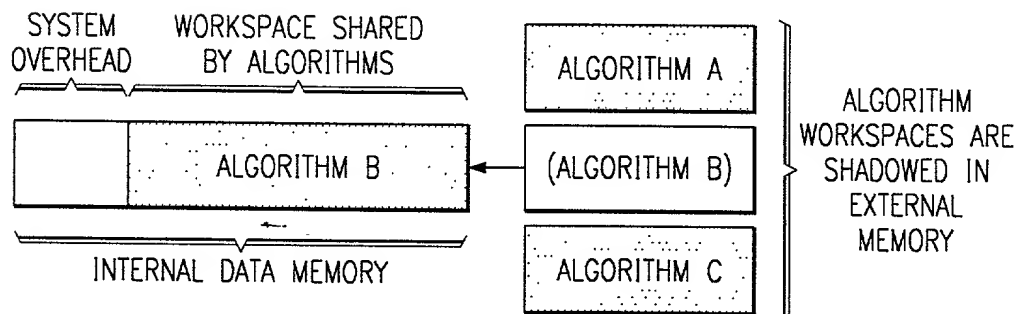
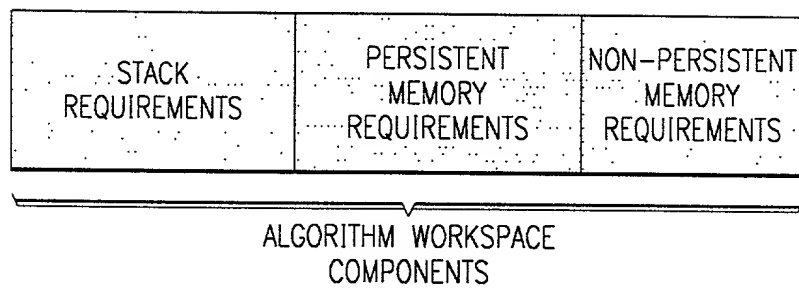
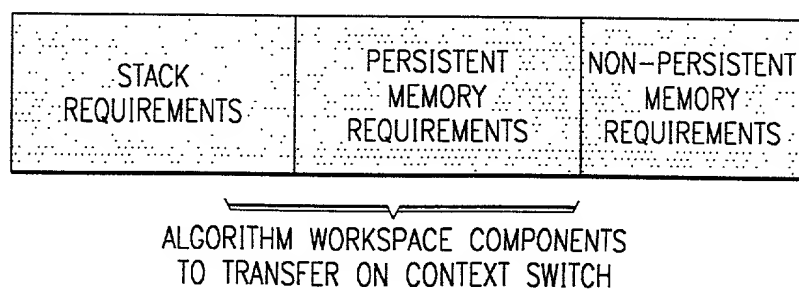


FIG. 16

FIG. 17



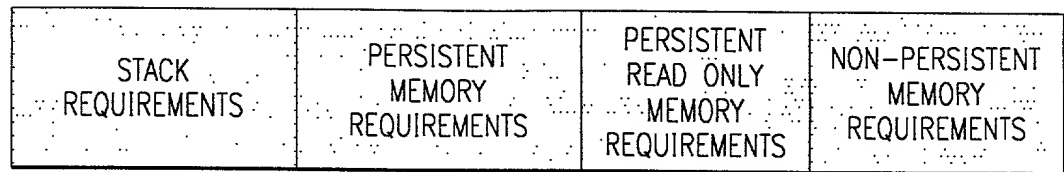
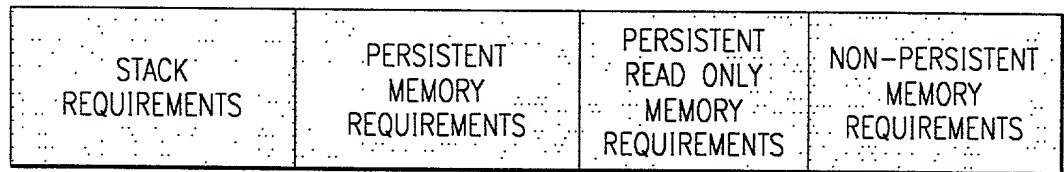
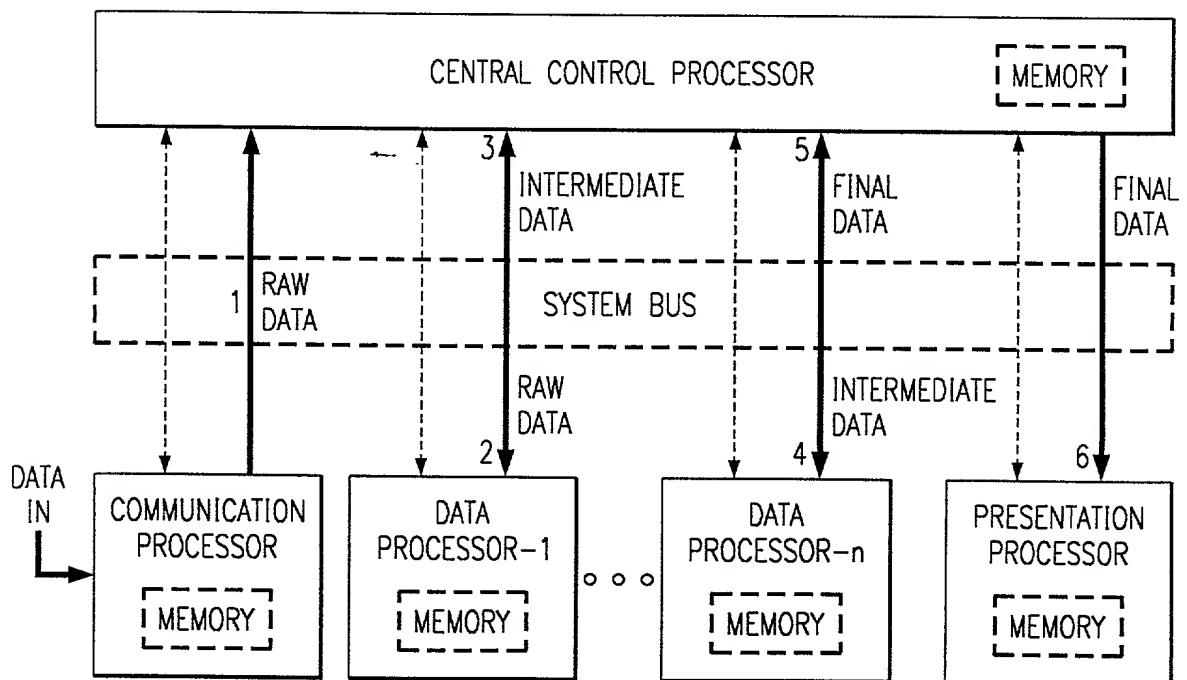


FIG. 18 ALGORITHM WORKSPACE COMPONENTS TO TRANSFER IN PRIOR TO ALGORITHM EXECUTION IF ALGORITHM REQUIRES CONSTANT TABLES (CONTEXT SWITCH IN ONLY)



READ ONLY PERSISTENT MEMORY DOES NOT NEED TO BE TRANSFERRED OUT ON CONTEXT SWITCH. THEREFORE ALGORITHM PAGE CHANGE-OUT IS MORE EFFICIENT.

FIG. 19



----- CONTROL PATHS

FIG. 20
(PRIOR ART)

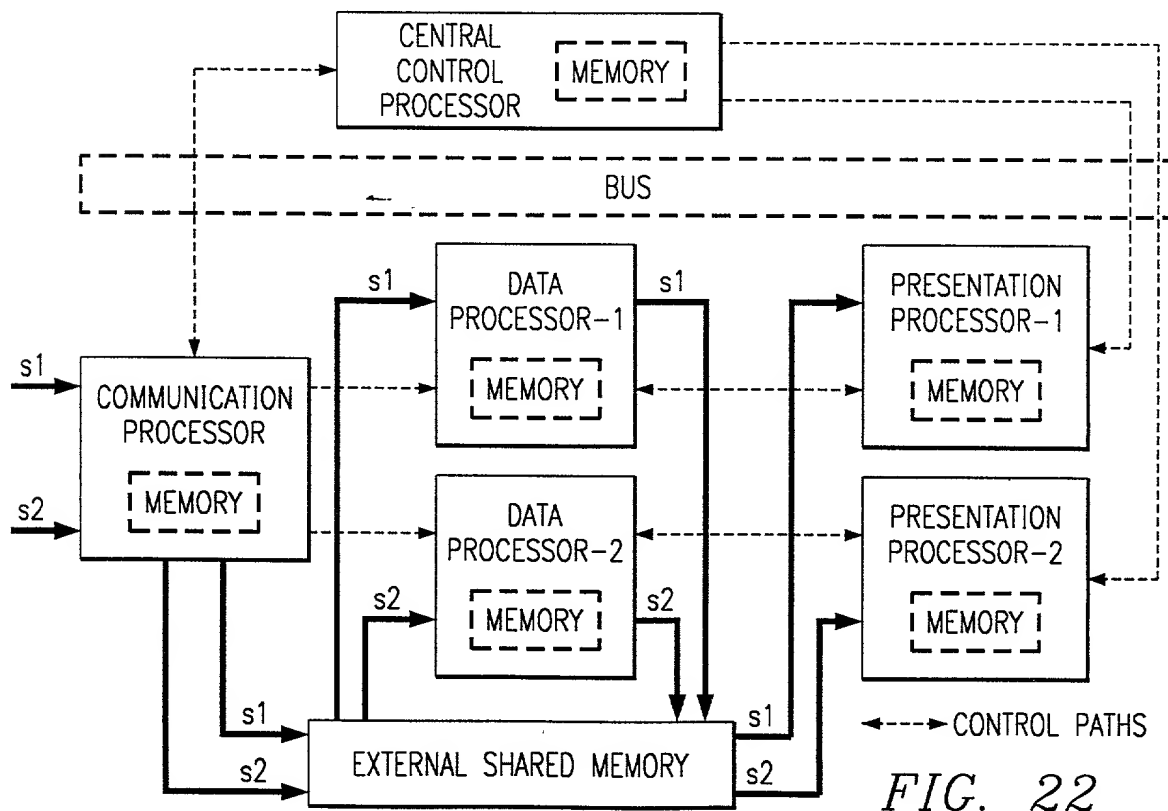
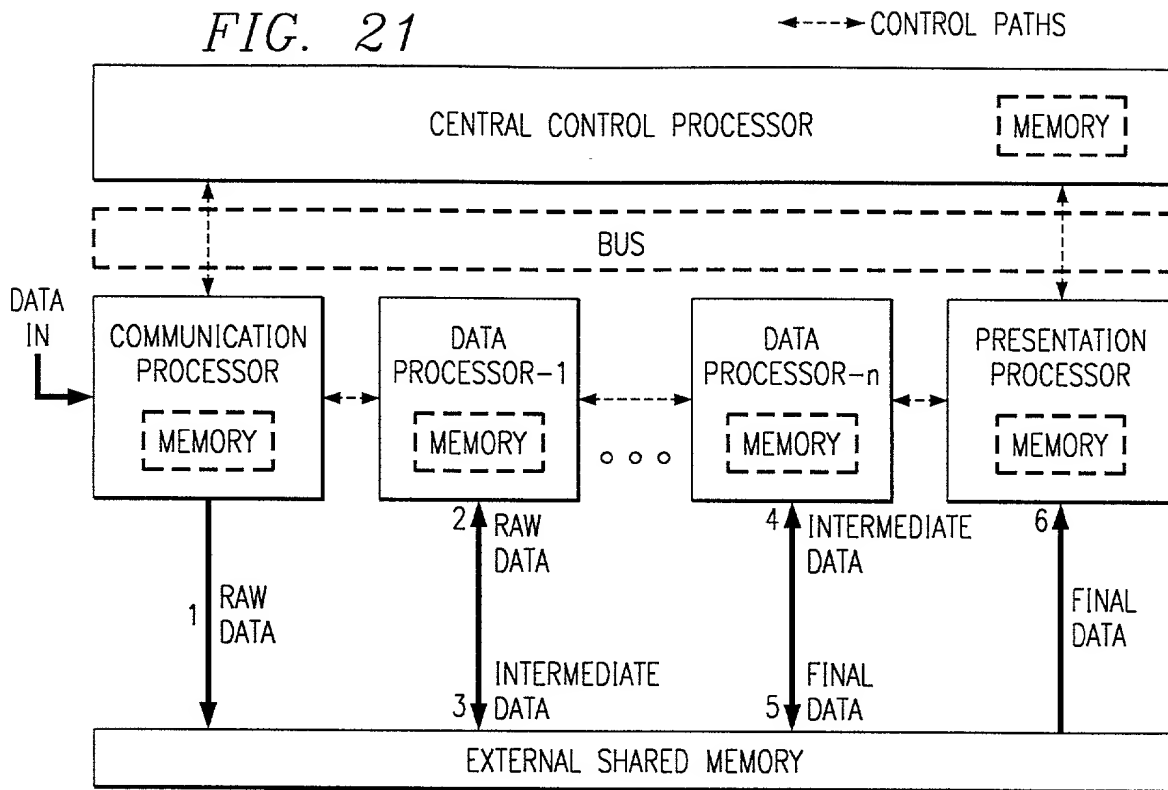
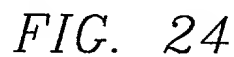


FIG. 22

[illegible]

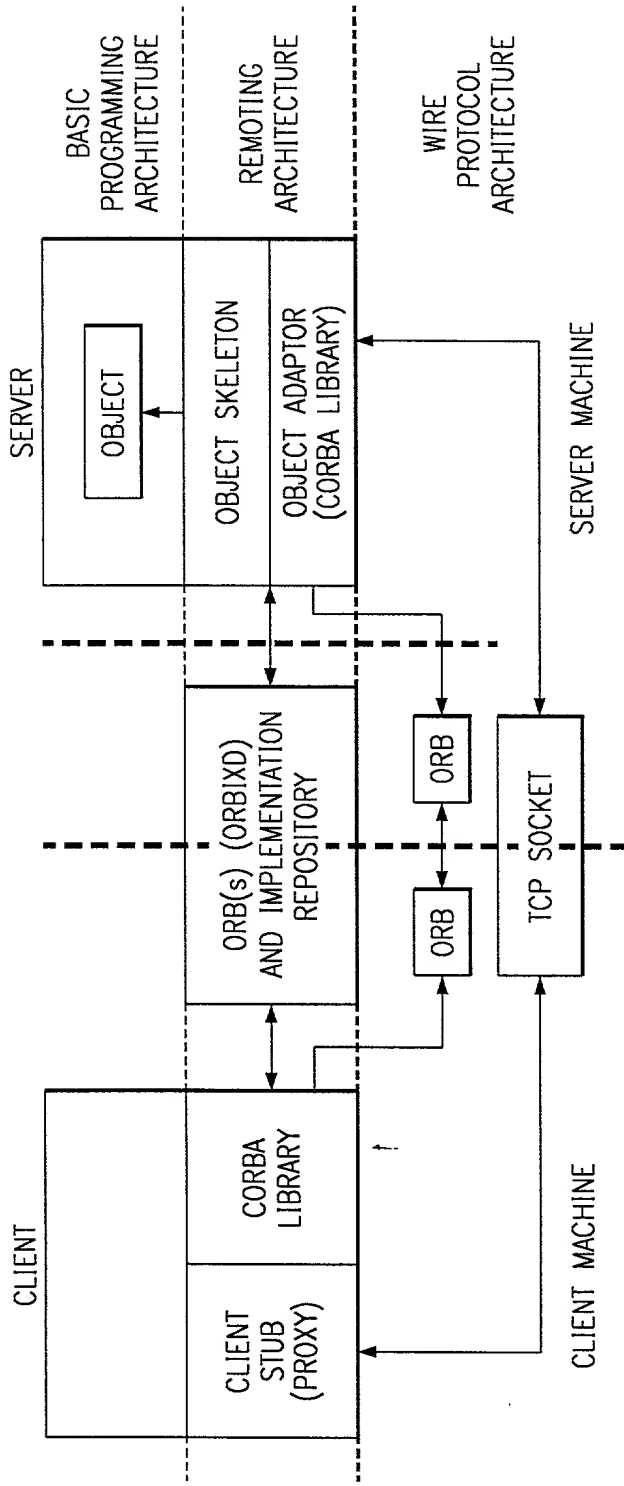


FIG. 25

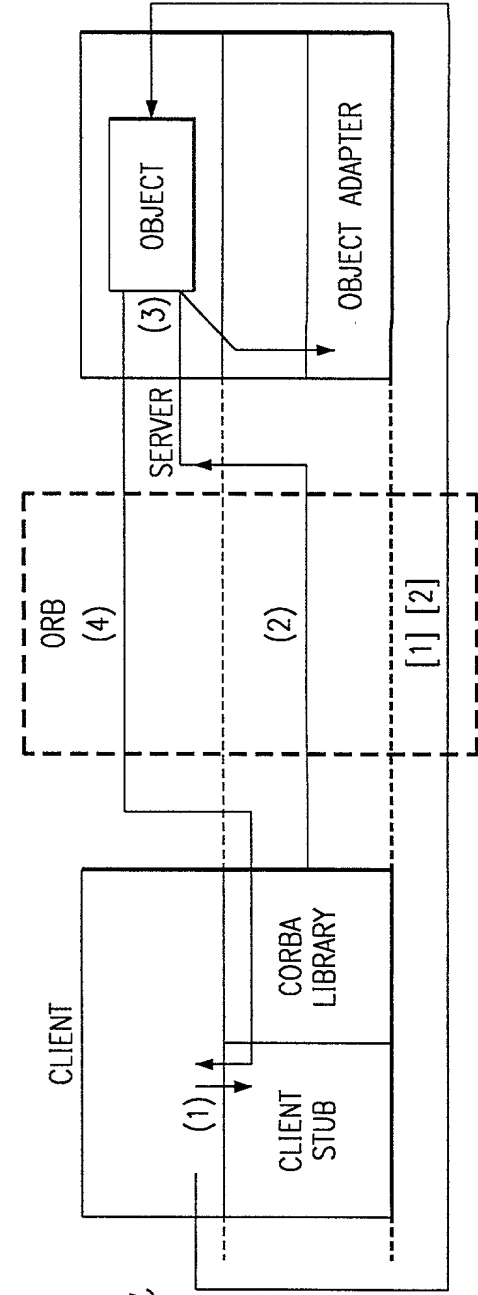


FIG. 26a

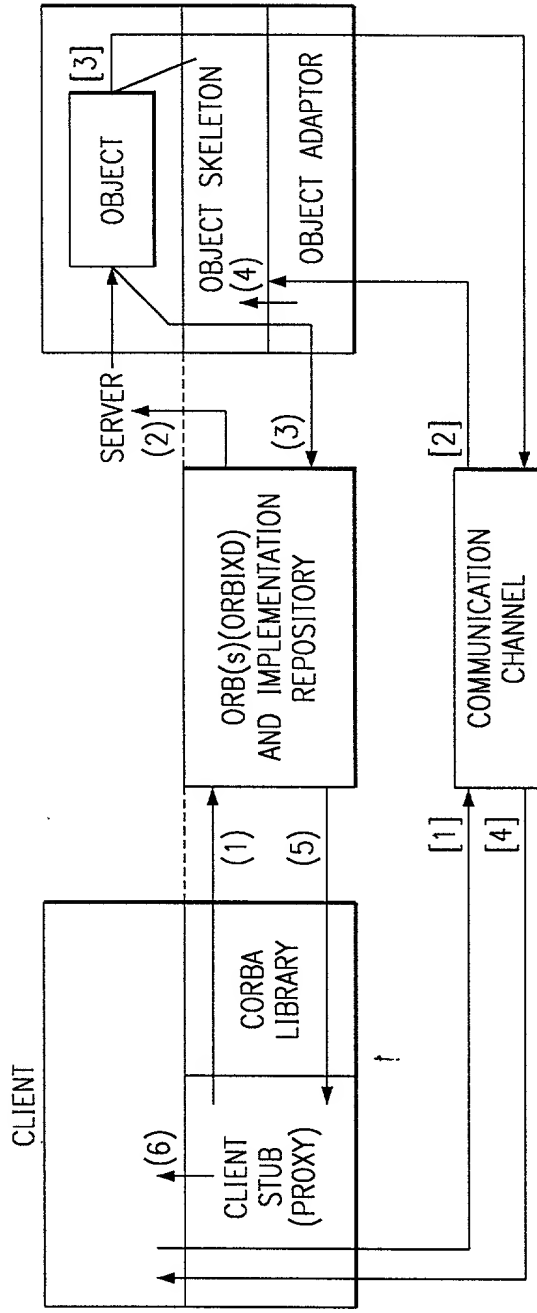


FIG. 26b

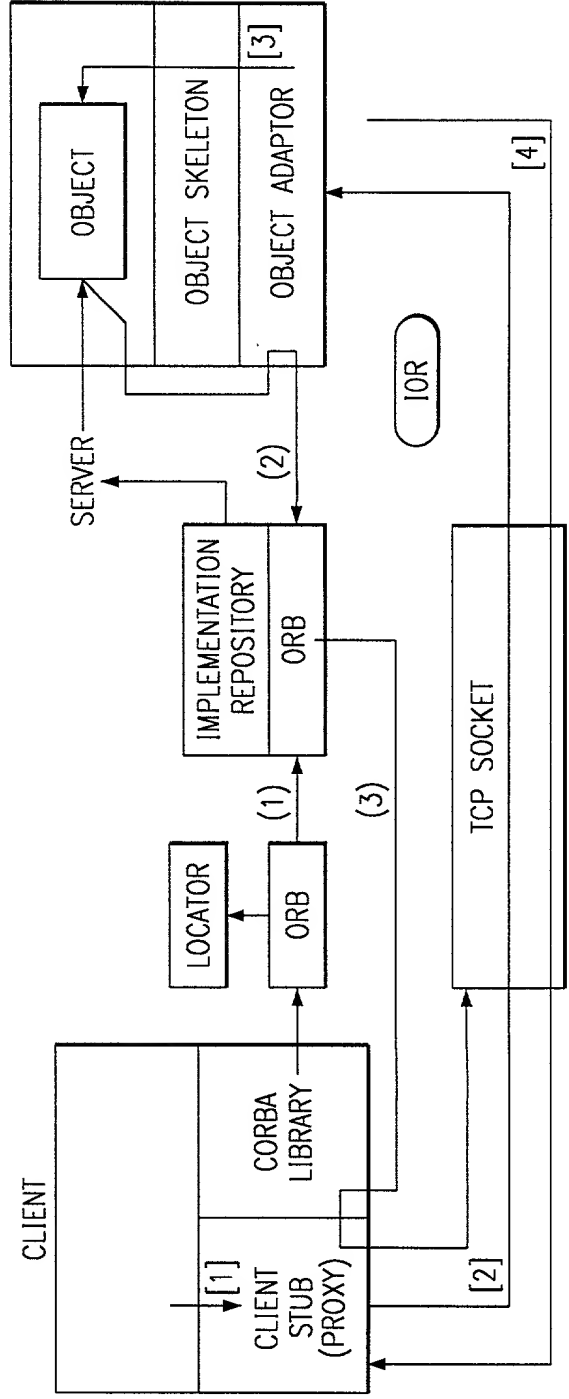


FIG. 26c

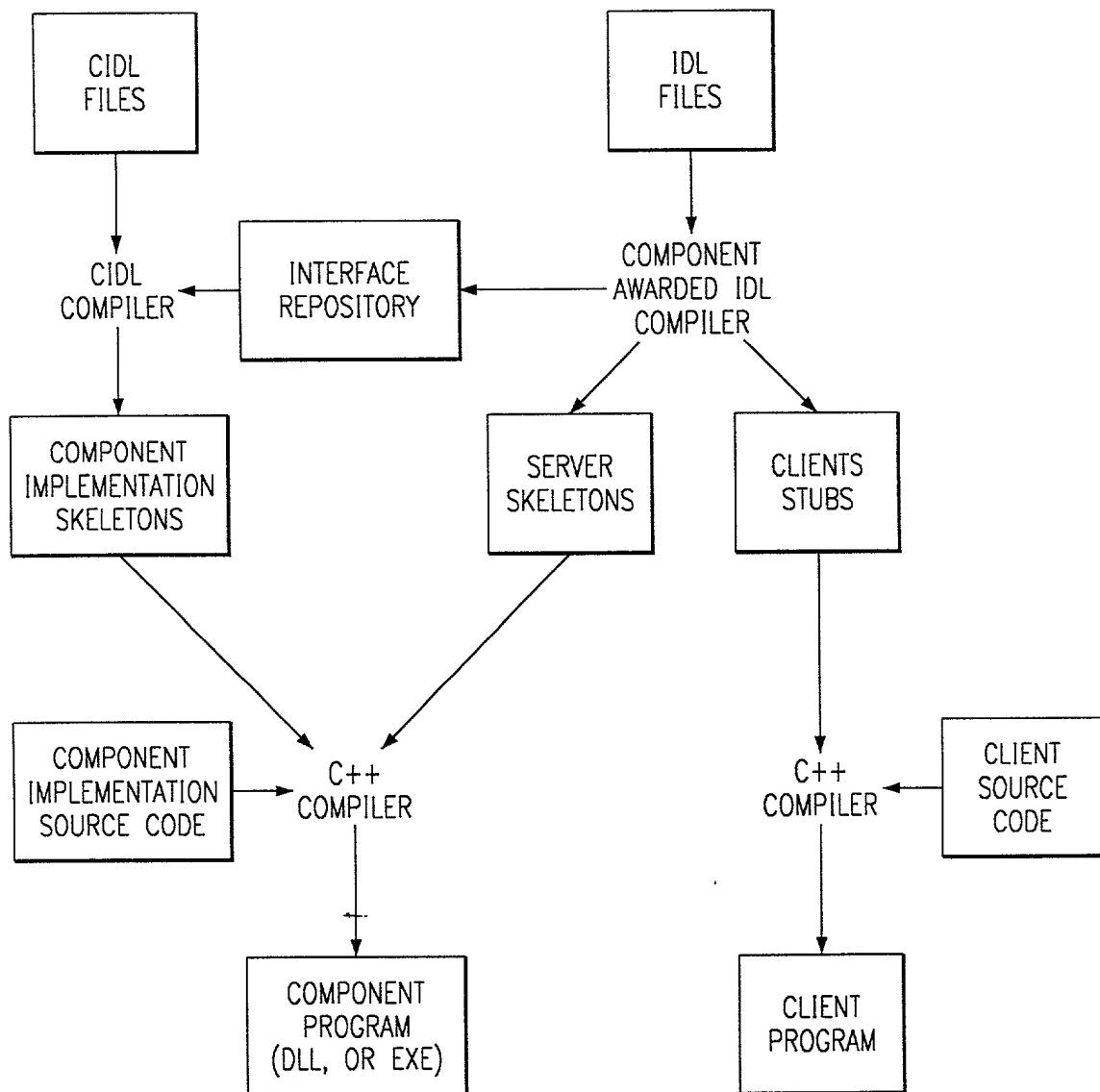


FIG. 27